REMARKS

Reconsideration of this application, as amended, is respectfully requested.

Claims 1-21 are pending. Claims 1-21 stand rejected.

Claims 1, 7, and 13 have been amended. Claims 22 and 23 have been added. Support for the amendments is found in the specification, the drawings, and in the claims as originally filed. Applicant submits that the amendments do not add new matter.

Rejections Under 35 U.S.C. § 102(e)

Claims 1-21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,356,960 of Jones, et al. ("Jones"). The Examiner stated that

> As per claims 1-2 and 7-8, Jones discloses the claimed invention including a method comprising: recognizing an occurrence of a user-specified event (recognizing control signals being transmitted by external debugging device; see Abstract or CPU executing an "event" instruction or decode special "event"; col. 8, lines 26-67; col. 9, lines 1-15; col. 11, lines 40-67); generating a signal to cease bus access (issuing instructions to cease fetching and enter the suspend state; col. 9, lines 15-26; col. 12, lines 12-18) in a configurable system on a chip (computer system on integrated chip; Abstract), upon the occurrence of the user-specified event, the configurable system on a chip integrating at lease a central processing unit (CPU0 12 and 13), an internal system bus (e.g. p-link 15 or pipeline) and a configurable logic (e.g. event logic 44); allowing completion of all pending bus transactions (causes a CPU to drain the execution pipelines; col. 9, lines 35-38; col. 10, lines 15-18); stopping the system clock (stops all CPU execution or CPU is suspended; col. 9, lines 23-26) such that the state of the hardware is held static (see col. 9, lines 50-56); and accessing the static state of the hardware through a debug port (debug port 30) [see col. 9, lines 50-55).

(p. 3, Office Action 2/11/04)

Jones discloses

> In this example each CPU 12 and 13 is arranged to execute an instruction sequence in conventional manner. The instruction set will include a plurality of conventional instructions for a microcomputer but this example also includes an instruction to send an "event". An "event" is an exceptional occurrence normally caused by circumstances external to a thread of instructions. Events can be used to have similar effect as an "interrupt" or "a synchronous trap". Events may be prioritised in that they can cause a change in the priority level at which the CPU executes. An event may be sent by execution of an event instruction although hardware in the form of the event logic 44 can

carry out the function of some events without the execution of instructions in a service or handler routine.

Events which originate from execution of an instruction by a CPU are caused by execution of the event instruction. This can be used to send an "event" to a CPU such as one or other of the CPU's 12 or 13 on the same chip or it may be used to send an event to a CPU on a different chip through an external connection. The CPU which executes the event instruction may also send an event to a further module connected to the P-link system 15. The event instruction has two 64 bit operands, the event number and the event operand. With regard to the event number 0-63, bit 15 is used to determine whether or not the event is a "special event". When bit 15 is set to 1, bits 0-14 are used to define the type of special event. Bits 16-63 of the event number are used to identify the destination address of the CPU or module to receive the special event. The types of special event are set out below:

Event Name	EN.CODE	EN.OPERAND	Function
EVENT.RUN	1	Ignored	Resumes execution from suspended state of the receiving CPU
EVENT.RESET	3	Ignored	Generate a reset event on the receiving CPU
EVENT.SUSPEND	5	Ignored	Suspends execution of the Receiving

(col. 8, lines 26-67) (Emphasis added)

EVENT.SET RESET.HANDLER	7	boot address	RESET.HANDLER .rarw. RESET SHADOW HANDLER RESET.HANDLER .rarw. boot address

These special events may be sent from one CPU 12 or 13 to the other or alternatively they may be sent through the debug port 30 from an external host to either of the CPU's 12 or 13 on chip. The "event" will be sent as a bit packet of the type previously described.

(Col. 9, lines 1-15) (Emphasis added)

Applicant respectfully submits that claim 1 is not anticipated by Jones under 35 U.S.C. 102§(e). Amended claim 1 includes the following limitations:

A method for diagnosing programmable hardware comprising:

recognizing an occurrence of a user-specified event;

generating a signal to cease bus access, in a configurable system on a chip, upon the occurrence of ~~a~~ the user-specified event, the configurable system on a chip integrating at least a central processing unit, an internal system bus, and a configurable logic, the signal generated by a breakpoint unit connected to the internal system bus, such that the breakpoint unit can be programmed through a debug port;

allowing completion of all pending bus transactions;

stopping the system clock such that the state of the hardware is held static; and

accessing the static state of the hardware through ~~a~~ the debug port.

(Amended claim 1) (emphasis added)

Applicants respectfully submit that Jones does not disclose a breakpoint unit connected to the bus that can be programmed through the debug port. This breakpoint unit is configured to generate one or more signals upon the occurrence of a breakpoint event. That is, the breakpoint unit can be programmed, through the debug port, with one or more events, the occurrence of which will cause the breakpoint unit to generate a clock signal that results in ceasing bus access.

Jones does not include this limitation. The special events of Jones cannot be programmed through the debug port. The "events" of Jones may be sent from either CPU or from an external host through the debug port, however, Jones does not disclose a breakpoint unit that may be programmed through the debug port.

For these reasons, applicant respectfully submits that amended claim 1 is not anticipated by Jones. Given that claims 2 – 6, 19, 22 and 23 depend, directly or indirectly from claim 1, applicant respectfully submits that claims 2 – 6, 19, 22 and 23 are, likewise, not anticipated by Jones. Further, given that amended claims 7 and 13 include the limitation of a breakpoint unit connected to the bus that can be programmed through the debug port, and given that claims 8 – 12 and 20 and claims 14 –18 and 21 depend directly or indirectly from claims 7 and 13, respectively, applicant respectfully submits that claims 7 – 18, 20 and 21 are, likewise, not anticipated by Jones

Applicant respectfully submits that claims 4 and 10 are not anticipated by Jones. Claim 4 includes the following limitations.

> The method of claim 1, wherein allowing completion of all pending bus transactions includes monitoring the bus for pending bus transactions.

(Claim 4) (Emphasis added)

Claim 10 includes the following limitation.

> The machine-readable medium of claim 7, wherein allowing completion of all pending bus transactions includes monitoring the bus for pending bus transactions.

(Claim 10) (Emphasis added)

In rejecting claims 4 and 10 as being anticipated by Jones, the Examiner has referenced Jones at column 9, lines 35 – 38. Jones discloses the following.

> Entering the suspend state causes a CPU to drain the execution pipelines. This takes an implementation defined period of time. While a CPU is suspended its execution context may be changed in any of the following ways.

(Col. 9, lines 35-38) (Emphasis added)

Jones does not disclose the limitation of monitoring the bus for pending bus transactions. Moreover, Jones indicates that allowing completion of pending bus transactions is accomplished by waiting "an implementation defined period of time."

For these reasons, applicant respectfully submits that claims 4 and 10 are not anticipated by Jones.

Applicant respectfully submits that claims 5 and 11 are not anticipated by Jones. Claim 5 includes the following limitations.

> The method of claim 4, wherein allowing completion of all pending bus transactions further includes generating a qualified clock freeze cycle upon completion of all pending bus transactions.

(Claim 5) (Emphasis added)

Claim 11 includes the following limitations.

> The machine-readable medium of claim 10, wherein allowing completion of all pending bus transactions further includes generating a qualified clock freeze cycle upon completion of all pending bus transactions.

(Claim 11) (Emphasis added)

In rejecting claims 5 and 11 as being anticipated by Jones, the Examiner has referenced Jones at column 9, lines 35 – 38 (see above).

Jones does not disclose the limitation of "generating a qualified clock freeze cycle upon completion of all pending bus transactions."

For this reason, applicant respectfully submits that claims 5 and 11 are not anticipated by Jones.

Applicant respectfully submits that new claim 22 is not anticipated by Jones. Claim 22 includes the following limitations.

> The method of claim 1 wherein the breakpoint unit is connected to a plurality of buses such that the breakpoint unit generates a signal in response to a user-specified event on any of the plurality of buses.

(New claim 22) (Emphasis added)

Jones does not disclose a plurality of buses connected to the event logic unit 44 that generates a signal on any of a plurality of buses. For this reason, applicant respectfully submits that claim 22 is not anticipated by Jones.

Applicant respectfully submits that new claim 23 is not anticipated by Jones. Claim 23 includes the following limitations.

> The method of claim 22 wherein the breakpoint unit is programmed to generate multiple signals upon the occurrence of a user-specified event.

(New claim 23) (Emphasis added)

Jones does not disclose the limitation of a breakpoint unit that is programmed to generate multiple signals upon the occurrence of a user-specified event. For this reason, applicant respectfully submits that claim 22 is not anticipated by Jones.

It is respectfully submitted that in view of the amendments and arguments set forth herein, the applicable rejections and objections have been overcome. If there are any additional charges, please charge Deposit Account No. 02-2666 for any fee deficiency that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 5/11/04 By: _____

Tom Van Zandt
Reg. No. 43,219

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(408) 720-8598